

United States Patent Application

of

K. Scott RAMEY,

Michel BURGER, and

Larry DAVID

for

**METHOD, APPARATUS, AND ARTICLE OF MANUFACTURE FOR
WEB-BASED CONTROL OF A UNIFIED MULTI-SERVICE COMMUNICATION
SYSTEM**

I. BACKGROUND OF THE INVENTION

A. Field of the Invention

The present invention relates generally to independently controlling the elements of a communications system and more particularly to methods, apparatus, and articles of manufacture for web-based control of a communications system.

B. Description of the Related Art

A legacy telephone system typically includes a number of telephone sets coupled to a private branch exchange (PBX) by physical wiring. A call server resides in the PBX and handles many different call-related functions for the telephone sets. Call control functions, service control functions, and user interface functions are typically controlled by the call server. Although several of those call-related functions can be initiated by users of the telephone sets, many are generally inaccessible. More specifically, while the call server can be controlled by a limited number of commands available to a user, many additional commands can only be issued by system administrators using restricted access interfaces.

Call control functions, such as call connection, call disconnection, and connection of the members of a conference call, are typically performed by a call server in response to a command from a telephone set coupled to the telephone system. For example, such a command could be the act of dialing a telephone number. Service control functions, such as establishing a conference call, performing last number redial, and initiating a voice mail function can be performed by a call server in accordance with a "policy" applied to a user's telephone. A system administrator controls the policy,

and thus controls which services are available to a user, by issuing commands to the call server through the restricted interface. Accordingly, users have a limited degree of control over the services available to them, which can typically be expanded only by contacting the telephone system administrator.

A need exists to expand the functionality of legacy telephone systems such that users thereof have greater control over their associated call server and telephone set. Thereby, users can access greater functionality without having to communicate requests to a telephone system administrator. Such a methodology increases the usefulness of legacy telephone systems by allowing them to be highly configurable by a user.

II. SUMMARY OF THE INVENTION

Methods, apparatus, and systems consistent with the principles of the present invention allow web-based control of the call control, service control and user interface functions of a legacy telephone system. The invention enables a legacy call server and a legacy telephony device to communicate with each other and with a web application. The invention uses the web application to control the legacy call server and the legacy telephony device, which perform the call control, service control and user interface functions of the telephone system. The web application also links the legacy call server and legacy telephony device to other web applications which supply data. A proxy server may be used to control access to the call server.

The invention may also provide a channel over which the web application and the call server communicate. The communications may include typical call server control commands, such as call control commands for controlling telephone calls or

service control commands for controlling telephone services. The invention may also translate data communicated between the call server and the web application.

The invention may also provide a channel over which the web application and the telephony device communicate. The invention may also translate data communicated between the telephony device and the web application. The invention may use an abstraction of the resources of a telephony device, or a class of several different telephony devices having similar characteristics. Web application data directed to abstract resources of a telephony device may be mapped to an actual resource of the device. The invention can also arbitrate access to a telephony device, and can route data from a telephony device to a specific web application. The invention may also use service plugins to provide an interface with web applications. Furthermore, the invention allows a web application to be the interface to another telephony device.

Another aspect of the invention provides a system for enabling a web application to control a legacy telephone system. The system may comprise a web application for controlling a legacy call server and a legacy telephony device, a call server wrapper for enabling the web application to communicate with the legacy call server, and a telephony device wrapper for enabling the web application to communicate with the legacy telephony device. The system allows a web application to control the call control functions and service control functions of the legacy call server and the user interface functions of the legacy telephony device.

It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as claimed.

III. BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and together with the description explain the advantages and principles of the invention.

FIG. 1 is a block diagram of a system for enabling a web application to control the elements of a legacy telephone system consistent with the principles of the present invention;

FIG. 2 is a flow chart of steps performed by a call server wrapper for enabling a web application to communicate with a call server system consistent with the principles of the present invention;

FIG. 3 is a flow chart of steps performed by a telephony device wrapper for enabling a web application to communicate with a telephony device consistent with the principles of the present invention;

FIG. 4 is a block diagram showing a logical configuration of software modules consistent with the principles of the present invention;

FIG. 5 is a diagram of a graphical user interface of a web application for communicating with a call server system; and

FIG. 6 is a diagram of a graphical user interface of a web application for communicating with a call server system.

IV. DETAILED DESCRIPTION

Reference will now be made in detail to an implementation of the invention as illustrated in the accompanying drawings. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. One skilled in the art, however, will realize that the present invention may be practiced without these specific details. In other instances, standard structures and devices are shown in block diagram form in order to facilitate description.

Sub
A1
The present application is related to, and incorporates by reference the U.S. Patent Application of K. Scott Ramey, Craig Will, and Larry David entitled METHOD, APPARATUS, AND ARTICLE OF MANUFACTURE FOR WEB-ENABLING LEGACY TELEPHONY DEVICES filed on October 8, 1999, attorney docket number 03384.0373-00000 and the U.S. Patent Application of K. Scott Ramey, and Michel Burger entitled METHOD, APPARATUS, AND ARTICLE OF MANUFACTURE FOR WEB-BASED CONTROL OF A CALL SERVER filed on October 8, 1999, attorney docket number 03384.0372-00000.

In one embodiment, steps according to the present invention are embodied in machine-executable software instructions, and the present invention is carried out in a processing system or systems by a processor executing the instructions, as will be described in greater detail below. Those skilled in the art will recognize that other embodiments, for example hardwired circuitry, may be used in place of, or in combination with, software instructions to implement the present invention.

Definitions

As used herein, a legacy telephone system can be a Private Branch Exchange (PBX) or other similar switching system, and the telephony devices associated with it. The legacy telephone system is controlled by a call server system which lacks the ability to communicate over an IP network, such as the World Wide Web, with a web application. As used herein, a legacy telephony device is primarily a voice communication device, such as a telephone, designed for use with a legacy telephone system and which lacks the ability to communicate directly with a web application. Apparatus and methods consistent with the principles of the invention interface an IP network and a legacy telephony system which lacks the ability to communicate with applications on the IP network.

As used herein, call server system is a collection of elements, including a call server, that function to control telephone calls and telephone services offered by the legacy telephone system or by a telephone system employing aspects of the present invention. As is known in the art, a call server can be implemented in many ways, for example as a computer program or application, or as hard-wired circuitry. Apparatus and methods consistent with the principles of the invention act as an interface between an IP network and a legacy call server system, which lacks the ability to communicate with web applications on an IP network.

Web application, as used herein, is generally a computer program or equivalent implementation in hardware, or a combination of both, that allows specified data to be accessed over an IP network, such as the World Wide Web. For example, a web application may provide the results of a directory lookup, voice messages, stock

quotations, or a streamed audio radio broadcast. A web application may also provide data in the form of commands, for example, commands to a call server system or a telephony device. A web application may connect directly to the present invention's application program interface (API), or it may have no inherent capability to communicate with the present invention.

Network Environment

FIG. 1 is a system diagram illustrating a networking environment consistent with the present invention. One embodiment of the present invention includes three components: a call server wrapper 167, which enables a legacy call server 165 to communicate with a web application, such as web application 110 or 180; a telephony device wrapper 140 which enables a legacy telephony device 135 to communicate with a web application, such as web application 110 or 180; and a web application, such as web application 110 or 180, which communicates with either or both call server 165 and telephony device 135. In one embodiment, call server wrapper 167 is a computer program or application executing on call server 165 and telephony device wrapper 140 is a telephony-device-specific computer program or application executing on a terminal proxy server (TPS) 145.

Although telephony device 135 is shown directly connected to an IP network 155, telephony device 135 cannot communicate directly with a web application such as web application 110 or 180. In the example shown in Fig. 1, telephony device 135 is a standard legacy telephone with the ability to communicate in internet protocol, such as a Nortel Model I2004 etherphone, but it has no knowledge of the API of any web applications, nor does a web application have any knowledge of it's API. Therefore,

they cannot communicate directly. Telephony device 135 communicates directly only with telephony device wrapper 140, which has full awareness of telephony device 135's API.

Wrappers 167 and 140 communicate over IP network 155 and provide API's to web applications connected to IP network 155 for communicating with call server 165 and telephony device 135 respectively. Entities shown connected to IP network 155 can access each other in a standard manner, for example using a universal resource locator (URL).

FIG. 1 depicts two web servers, a web server 105, and a web server 185 communicatively connected to IP network 155. Web server 105 and web server 185 make a "411" directory web application 110 and a directory glue application 180 available, respectively, over IP network 155. Directory glue application 180 provides an interface between 411 directory web application 110 and wrappers 167 and 140, which allows 411 directory web application 110 and wrapper programs 167 and 140 to communicate with each other. Directory glue application 180 may also provide a user interface, such as a graphical user interface (GUI), for communicating with 411 directory web application 110, wrapper programs 167 and 140, and ultimately with legacy telephony device 135 and legacy call server 165. In one embodiment, directory glue application 180 is a remotely located part of wrappers 140 and 167. 411 directory web application 110 is merely one example of a web application that the present invention enables to inter-work with a legacy telephone system. One of ordinary skill in the art will realize that the present invention enables inter-working with any web application.

In one embodiment, applications, such as 411 directory web application 110 and directory glue application 180 are Java applets executable on a web browser. For example, PC 125 loads a web page including a Java applet web application from a web application server, such as web server 105 or web server 185. The Java applet web application allows control of call server 165 and telephony device 135 from the browser of PC 125. The web application need not be located remotely from the web browser; it could, for instance, be hosted on PC 125.

A user desktop 130 comprises PC 125 and telephony device 135. In one embodiment, telephony device 135 is a conventional legacy telephone and PC 125 a conventional PC. Telephony device 135 is conventionally connected to PBX 160 through an IP network 155.

TPS 145 is a conventional high-speed server computer, such as a workstation. TPS 145 is connected to IP network 155. In this embodiment, TPS 145 provides a host for telephony device wrapper 140, which provides a communication channel between telephony device 135 and web entities such as 411 directory web application 110.

In one embodiment of the present invention, a user proxy server 115 is communicatively connected to IP network 155. User proxy server 115 controls access to call server wrapper 167 and call server 165. In this embodiment, a web application, such as 411 directory web application 110 and directory glue application 180, communicates with user proxy server 115 instead of call server wrapper 167 and telephony device wrapper 140. User proxy server 115 determines the identity and privileges of a web application user, and determines whether to pass data from the web

application to call server wrapper 167 and telephony device wrapper 140 based on the user's identity and privileges.

Call server 165 controls calls and services in PBX 160. PBX 160 is connected to PSTN 170, which is connected to telephony device 175. As is well-known in the art, call server 165 controls the calls and services in PBX 160. Telephony device 135 is connected to PBX 160 and can make and receive calls in a conventional manner; the presence of the present invention is transparent to a user when telephony device 135 is used in a conventional manner. In one embodiment, PBX 160 comprises a conventional legacy PBX, which thus lacks the ability to communicate with web applications on IP network 155, which could be, for example, the World Wide Web.

Directory Lookup

Assume for the purposes of example that telephony device 135 has a two-line LCD display, a speaker, and a handsfree mode. A user at telephone 135 wishing to call a person at telephone 175 needs to know the telephone number of telephone 175. In accordance with the principles of the present invention, a user uses 411 directory web application 110 to place the call, as will be described below.

In one embodiment of the present invention, a user uses telephone 135 to look up the telephone number by accessing 411 directory web application 110 through telephony device wrapper 140. 411 directory web application 110 and telephony device wrapper 140 allow the user to search the directory using the user interface available on telephone 135. For example, the directory may display two lines at a time on the two line display of telephone 135, allowing the user to jump and scroll through the directory using the function keypad of telephone 135. Once the correct name is found, the user

may select the corresponding number by, for example, pressing a specified keypad button. Telephony device wrapper 140 forwards the selected number to 411 directory web application 110.

In response to the selection, 411 directory web application 110 communicates with telephone 135 through telephony device wrapper 140, commands telephone 135 into handsfree mode, and activates the telephone's speaker. 411 directory web application 110 also communicates with call server 165 through call server wrapper 167, and commands call server 165 to make a call connection from telephone 135 to the telephone number selected by the user.

The user hears telephone 135 go offhook, and then hears destination telephone 175 ring or return a busy signal as the connection is made. Thus, the user hears exactly what would be heard if a handsfree call had been made by manually dialing with the keypad.

In another embodiment of the present invention, web application 110 does not communicate directly with call server 165 through call server wrapper 167. Instead, it communicates with user proxy server 115, and user proxy server 115 communicates with call server 165 through call server wrapper 167. This embodiment provides enhanced security because call server wrapper 167 need only accept commands from a particular number of known user proxy servers 115, instead of numerous telephony devices 135.

Complimentary User Interface

Assume for purposes of example that a user at user desktop 130 wishes to call the person at telephone 175, but does not know the telephone number. Assume also

that 411 directory web application 110 is a legacy web application having no inherent ability to inter-work with call server wrapper 167 and telephony device wrapper 140. Further assume that telephony device 135 is a telephone featuring a speaker and a handsfree mode, and that PC 125 is associated with telephony device 135; for example, it knows the IP address and telephone number of telephony device 135.

To make the call in a conventional architecture, the user accesses 411 directory web application 110 using PC 125 and searches the directory for the name of the person at telephone 175. The user then presses the handsfree key on telephone 135 to go offhook, waits for a dialtone, and manually enters on the dial keypad the telephone number displayed by the GUI of 411 directory web application 110. At this point, the user hears destination telephone 175 ring or return a busy signal.

The present invention implemented using a web browser on PC 125 allows the user to place the same call without using the user interface of telephone 135. To place the call, the user first downloads directory glue application 180 into PC 125 from web server 185 using a web browser on PC 125. Directory glue application 180 is specifically designed to place a call and to communicate with 411 directory web application 110, call server wrapper 167, and telephony device wrapper 140. Directory glue application 180 runs on the web browser and provides a GUI on PC 125 for presentation and control of the functions of directory glue application 180.

Using the GUI on PC 125, the user searches a directory provided by 411 directory web application 110. Directory glue application 180 presents the telephone number associated with each name in a way that makes the number selectable, for example, as a hyperlink. The user finds the name of the person at telephone 175 and

selects the person's telephone number to make a call, for example, by clicking on the hyperlink.

In response to the user's selection, directory glue application 180 executing on PC 125's web browser initiates a call. Directory glue application 180 gets the information identifying telephone 135 from PC 125, and using call server wrapper 167's API, commands call server 165 to make a call connection from telephone 135 to the telephone number selected from 411 directory web application 110.

Call server wrapper 167 translates and passes the call control command and associated data to call server 165, and translates and passes call server 165's response, for example, that the call is proceeding, back to directory glue application 180. Directory glue application 180 then updates its browser GUI with the response information.

Using a standard device control interface (not shown), call server 165 communicates over IP network 155 with telephony device wrapper 140. Telephony device wrapper 140 is designed to communicate with call server 165's device control interface. Telephony device wrapper 140 may be designed to communicate with applications in addition to the device control interface of call server 165, for example by using a plugin designed to communicate with a web application. Call server 165 sends a command to turn on the handsfree mode of telephone 135 and sends appropriate progress data. Telephony device wrapper 140 translates the command and data to a format telephone 135 can understand, for example H.248 command format, and transmits them to telephone 135. In response, telephone 135 goes into handsfree mode, and the user hears telephone 135 go offhook and hears destination telephone

175 ring or return a busy signal as the connection is made, exactly as if the user had made a handsfree call by manually dialing with the keypad.

Once the call is proceeding, it can be controlled in a conventional manner via the user interface of telephone 135. In one embodiment of the present invention, the call can also be controlled by directory glue application 180. For example, directory glue application 180 monitors the call state through call server wrapper 167 as the call proceeds. Directory glue application 180 presents a GUI control, for example, a "hangup" button, for controlling the ongoing call. If the user clicks on the hangup button, directory glue application 180 sends the appropriate commands to release the call to call server 165 through call server wrapper 167.

In another embodiment, 411 directory web application 110 is designed with the ability to inter-work with the present invention embodied in call server wrapper 167 and telephony device wrapper 140, obviating the need for directory glue application 180.

The above complimentary user interface example can also be used to illustrate the functioning of the present invention embodied with a user proxy server 115. In the user proxy server embodiment, all direct communications to entities connected to IP network 155 are originated by user proxy server 115. For purposes of example, in addition to the assumptions above, further assume that telephone 135 was previously registered with user proxy server 115, for example, by automatic self registration when telephony device wrapper 140 was initiated, or manual registration by a user at desktop 130. Thus, user proxy server 115 is aware of how to communicate with telephone 135 through telephony device wrapper 140.

To do a directory lookup, a user at PC 125 first registers with user proxy server 115 using a web browser. Then the user performs a directory lookup using 411 directory web application 110 through user proxy server 115; that is, the user communicates with user proxy server 115, which in turn communicates with web server 105. The response of the directory lookup is received by user proxy server 115, which sends a representation of the response to the user at web browser 125, or at telephone 135 through telephony device wrapper 140. The user can make a call based on the response information, for example the telephone number of the person looked up, by using either pc-based web browser 125, or telephone 135.

In this embodiment, however, the request to make a telephone call is transmitted from PC 125 or telephone 135 to user proxy server 115, which in turn communicates an appropriate connect command to call server wrapper 167. When user proxy server 115 receives a response from call server 165, it may command telephone 135, through telephony device wrapper 140, to go off hook and open a media channel for the call.

Email A Telephone Call

A user at user desktop 130 can email a telephone call to a person at another telephone using the present invention. Assume for the purposes of this example that a user at user desktop 130 wishes to email a message, including instructions to call back and confirm, to a recipient. Further assume that PC 125 is associated with telephony device 135; for example, that it knows the IP address and telephone number of telephony device 135, and that the recipient has the same desktop configuration (not shown) as desktop 130. Finally, assume that a mailing web application (not shown) is made available over IP network 155 by web server 185.

To email a telephone call, the user at desktop 130 loads the mailing web application onto a web browser executing on PC 125 from web server 185. The mailing web application allows the user to include a method for activating a telephone call in the email message, for example, by using a hyperlink or appending a Java applet. Using the mailing web application GUI, the user creates an email message that concludes, for example, the hyperlink "Please call be back to confirm at <http:\abcwww\cgi-bin\callme>," and sends the message to the recipient.

The recipient opens the email message and double clicks on the hyperlink to activate a telephone call to the user. The hyperlink goes to an IP network address and executes a CGI script on the server for the address. The CGI script activates the mailing web application, which places the call by sending appropriate commands to call server 165 through call server wrapper 167.

Call server 165 causes the recipient's telephone (not shown) to go offhook, then rings the user's telephone 135. When the user picks up, the call is established.

The Call Server Wrapper Program

In order for a web application and a call server to communicate with each other, they must either use the same API format or have a translator between API's. Apparatus and methods consistent with the principles of the present invention translate data transferred between a web application and a call server system into formats compatible with each.

FIG. 2 is a flow chart illustrating the steps performed by call server wrapper program 167 of Fig. 1 consistent with the principles of the present invention. Call server wrapper program 167 first determines whether data was received from a web

application (step 215). If web application data was received, call server wrapper program 167 translates the data from web application format to call server system format (step 220), and transmits the translated data to the call server system element handling the communication from the web application (step 225). Web application format is a format that call server wrapper program 167 can receive and understand, which is used by a web application communicating with call server wrapper program 167. Call server system format is a format that a call server system can receive and understand. Processing then returns to determining whether additional web application data was received (step 215).

If web application data is not received (step 215), call server wrapper program 167 determines whether call server data was received from the call server system element handling the communication from the web application (step 230). If call server data was received, call server wrapper program 167 translates the data from call server system format to web application format (step 235), and transmits the translated data to the web application (step 240). Processing then returns to determining whether additional web application data was received (step 215).

Call server wrapper program 167 handles translations between data in a web application format and data in a call server format. This allows legacy call server systems to interact with web-enabled entities on an IP network, such as the World Wide Web.

Call Server Wrapper - Web Application Interface

Call server wrapper program 167 comprises two interfaces: a web application program interface (API) through which it communicates with a web application

executing on PC 125, and a call server system interface through which it communicates with an entity in call server 165. Call server wrapper program 167 provides a uniform user interface for all types of call server systems, such as PBX's, and may be designed to interface with any model and manufacturer. Thus, call server wrapper program 167 allows web applications and call server systems to interact with each other without reprogramming either end to provide data in a format specific to the other end.

The web API of call server wrapper program 167 defines the web application data format. That is, the web API specifies the format of the data communicated between a web application and call server wrapper program 167. In one embodiment, the web API allows the web application data to be in the format of a call control command. A call control command causes call server 165 to perform a function which controls a telephone call or which returns the status of a telephone call or of call server 165. For example, a call control command may cause call server 165 to make a connection between two telephony devices.

In one embodiment, call server control commands are broken down into primitives. The number of primitives may be fewer than the number of call server control commands. The web API specifies that the web application data be in the form of the primitives. Reducing the number of call control commands to a small group of primitives simplifies the call server wrapper interface to web applications. Complex call control functions are realized by using combinations of the call control command primitives.

In one embodiment, the web API allows the web application data be in the format of a service control command. A service control command causes call server

165 to affect a telephony service for a telephony device. For example, the web API may include an add caller id(user) service control command, which enables a caller id service for the telephone identified by the user parameter. In another example, the web API includes a ring(user, ring) service control command, which changes the ring pattern according to the ring parameter for the telephone identified by the user parameter.

Call Server Wrapper - Call Server System Interface

The call server system that call server wrapper 167 communicates with defines the call server data format. More particularly, the call server system's interface defines the format requirement of the data communicated between call server 165 and call server wrapper program 167. The present invention converts data from a web application format into a call server system data format before transmitting the converted data to the call server system.

Many conventional call server systems have a computer telephony interface (CTI) in place for outside control of a call server. In one embodiment, call server wrapper 167 communicates with call server 165 in a call server data format defined by the CTI.

The Telephony Device Wrapper Program

To communicate, a web application such as 411 directory web application 110 and telephony device 135 must have a communication path between them and use the same API or have a translator. Apparatus and methods consistent with the principles of the present invention provide a communication channel between a web application and

a telephony device and translate data transferred between web application and a telephony device into formats the APIs of each understands.

Figure 3 is a flow chart of the steps performed by telephony device wrapper program 140 consistent with the principles of the present invention. Telephony device wrapper program 140 first determines whether any data was received from a web application, for example 411 directory web application 110 (step 315). If web application data was received, telephony device wrapper program 140 translates the data from a web API format to a telephony device data format (step 320) and transmits the translated data to telephony device 135 (step 325). Web API format is a format that telephony device wrapper program 140 can receive and understand, which must be used by a web application communicating with telephony device wrapper program 140. Telephony device data format is a format in which the telephony device 135 communicates.

If no web application data has been received (step 315), telephony device wrapper program 140 determines whether telephony device data was received from telephony device 135 (step 330). If telephony device data was received, telephony device wrapper program 140 translates the data from telephony device data format to web API format (step 335) and transmits the translated data to 411 directory web application 110 (step 340). Processing then continues with telephony device wrapper program 140 determining whether web application data has been received (step 315).

Logical Configuration

One embodiment of telephony device wrapper is composed of logical components as shown in Fig. 4. More particularly, the embodiment is composed of a

device driver 420, a context manager 410, and service plugin 430. The software configuration of Fig. 4 is only one possible embodiment of elements consistent with the principles of the present invention.

Telephony Device Wrapper - Telephony Device Interface

Device driver 420 communicates with telephony device 425 using a telephony device data format. Telephony device 425 defines the requirements of the telephony device data format. More particularly, the interface for telephony device 425 defines the format requirements of the data communicated between telephony device 425 and device driver 420. Device driver 420 converts web application data from context manager 410 into a telephony device data format before transmitting the converted data to telephony device 425. Conversely, device driver 420 converts telephony device data from telephony device 425 into a web API format before passing the converted data to context manager 410.

The telephony device data formats used by existing telephony devices are well-known in the art. Generally, telephony devices use a well-defined communications protocol such as H.248. In one embodiment, context manager 410 communicates directly with telephony device 425 using the device's native data format.

Telephony Device Wrapper - Web Application Program Interface

The present invention also provides a simple user interface for each telephony device, so that a wide variety of web applications can easily access a device without knowledge of the telephony device's specific operating details. Context manager 410

contains a telephony device wrapper web application program interface (API) 415 for communicating, either directly or indirectly, with a web application.

Telephony device wrapper web API 415 specifies the format of the data communicated between a web application, such as web application 445, and the present invention. In another embodiment, if a web application such as remote web application 432 cannot communicate with the telephony device wrapper web API 415 in web API data format, then a service plugin 430 translates the native data format of web application 432 into the appropriate web API data format before passing the web application's data to telephony device wrapper web API 415.

In one embodiment, wrapper web API 415 uses an abstraction to represent telephony device 425. The abstraction generalizes the characteristics of a telephony device resource so that a web application can use a resource without knowing specifics of the resource's implementation. A resource of a telephony device is any feature or ability of the device that can be used by a web application. For example, a resource could be a handset, a speaker, a two-line LCD display, or keypad buttons. An example of an abstraction of a resource is to represent a two-line, 28 character, LCD display as a general text display resource with no limits on the number of lines or characters.

In another embodiment, the wrapper web API 415 uses an abstraction to represent a class of telephony devices which share similar characteristics. For example, a class of display telephones may include member types that have two line displays, four line displays, and four inch by four inch bit-mapped displays. All of these similar displays could be represented by an abstraction which defines a general text display resource with no limits on the number of lines or characters. Wrapper web API

415 may use this abstraction to communicate with web applications, hiding the details of the actual device display from the web application.

Context Manager

Although an abstraction provides a relatively simple API to web applications, the data received through an abstracted API generally cannot be directly used by a telephony device. In another embodiment, to alleviate this problem, context manager 410 maps the abstract data to the telephony device's actual resources. Mapping involves configuring the data transferred by remote web application such as remote web application 445, into a form that can be utilized by the telephony device 425. In an embodiment where telephony device wrapper web API 415 uses an abstraction to represent a class of telephony devices, the invention keeps track of which devices among those in the class are connected, and maps web application data according to each device's actual resources.

Telephony device 425 may have several resources. For example, telephony device 425 may have a speaker resource, a two-line display resource, a numeric keypad resource, and a function keypad resource. In one embodiment, context manager 410 arbitrates access to a telephony device's resources so that two or more web applications may share the resources of a single telephony device. To arbitrate, context manager 410 keeps track of the state of each web application and telephony device resource and prioritizes which web application may use a resource. For example, web application 445 may be using telephony device 425's two-line display, when web application 432 tries to write data to the same display. The context manager

410 arbitrates access to the two-line display, holding the web application 432's data until the first web application finishes using the display.

For data moving from telephony device 425 to a web application, one embodiment of the present invention routes signals, such as stimulus data coming from a telephony device to the web application. In this embodiment, context manager 410 keeps track of which stimulus resources of a telephony device belong to which web application, and directs stimulus inputs to web applications accordingly. For example, assume wrapper web API 415 for telephony device 425 contains a speaker resource and a display resource. Web application 445 uses the display resource to show a string of real time stock quotes requested by a user. At the same time, web application 432 uses the speaker resource to stream broadcast news audio. If, for example, the stock quotation web application 445 recognizes telephony device 425 keypad strokes "#9" as a command to stop providing quotations, and the telephony device user presses "#9," then the context manager 410 routes the "#9" stimulus data to the stock quotation web application 445 rather than to the streaming audio web application 432.

Service Plugins

In another embodiment of the invention, where a web application such as web application 432 cannot communicate with the present invention's telephony device wrapper web API 415 in web API data format, then the present invention provides a service plugin 430 between wrapper web API 415 and web application 432. Service plugin 430 allows web application 432 and the telephony device wrapper web API 415 to communicate with each other. Service plugin 430 is a telephony-device-specific and web-application-specific custom interface. For example, assume web application 432 is

designed to stream audio to a sound card on the PC hosting the web browser which accessed the web application. Web application 432 does not have information regarding communication with the present invention's wrapper web API 415, which includes, for example, an abstraction of a telephone audio speaker. To stream audio data from web application 432 to the telephone's speaker, service plugin 430 receives the setup control data transmitted by web application 432, converts it into API data format, and sends it to telephony device wrapper web API 415 for preparing the speaker of telephony device 425 to accept a streamed audio connection.

In another embodiment, a service plugin 430 contains a service user interface 435 which provides a telephony device-specific user interface for a web application 432.

In yet another embodiment, service plugin 430 is implemented as an optional plugin to the present invention. The optional plugin is only included if a user desires telephony device 425 to communicate with the specific web application for which the service plugin is customized. In this embodiment, the telephony device wrapper provides an execution environment for service plugin 430. An execution environment allows a software application to function on a host system. For example, the present invention may provide a JAVA™ virtual machine for executing service plugins implemented as JAVA™ plugins. A service plugin may be implemented to execute on a remote system and communicate with API 415 over a network, without loss of compatibility with the present invention. In another embodiment, service plugins may interact with each other, as well as with wrapper API 415.

Web Application Program

As previously mentioned, a web application is any entity on IP network 155 capable of providing data to either or both call server wrapper 167 and telephony device wrapper 140. In one embodiment, a web application communicates with a wrapper program using a web application data format.

Fig. 5 shows an example of a web browser graphical user interface 530 on PC 125 for a web application. A title 510 shows that a user at telephone number 555-1111 has loaded a directory lookup web application, such as 411 directory web application 110 on web browser 125. The web application's GUI allows the user to select a telephone number 515 from the directory list, using the cursor 520. If the user clicks on the selected number, the web application causes a call to be placed (denoted in area 525) by communicating with call server 165 through call server wrapper program 167. Web application 110 directs call server 165 to connect the telephone 135 at 555-1111 to the selected telephone 175 at 555-2222. In one embodiment, web application 110 communicates call control commands directing call server 165 to place the call. In another embodiment, web application 110 commands caller telephone 135 to go offhook and into handsfree mode when a connection is made.

Fig. 6 shows an example of a web browser graphical user interface 630 on PC 125 for a web application. A title 610 shows that a user at telephone number 555-1111 represented by telephony device 135 is invoking a telephone services web application (not shown) on web browser 125. The web application GUI allows a user to add or remove services for their telephone (denoted in area 625). The web application lists the services available 615 and allows the user to add or remove services by clicking on a checkbox 620 using the cursor 622. If the user clicks a checkbox, the web application

communicates this change to call server 165 through call server wrapper program 167.

The web application directs call server 165 to add the selected service, for example speed dialing, to the user's telephone 135 at 555-1111. In one embodiment, the web application communicates service control commands to cause call server 165 to add or remove the selected service.

Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the invention being indicated by the following claims.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2